FIG.1

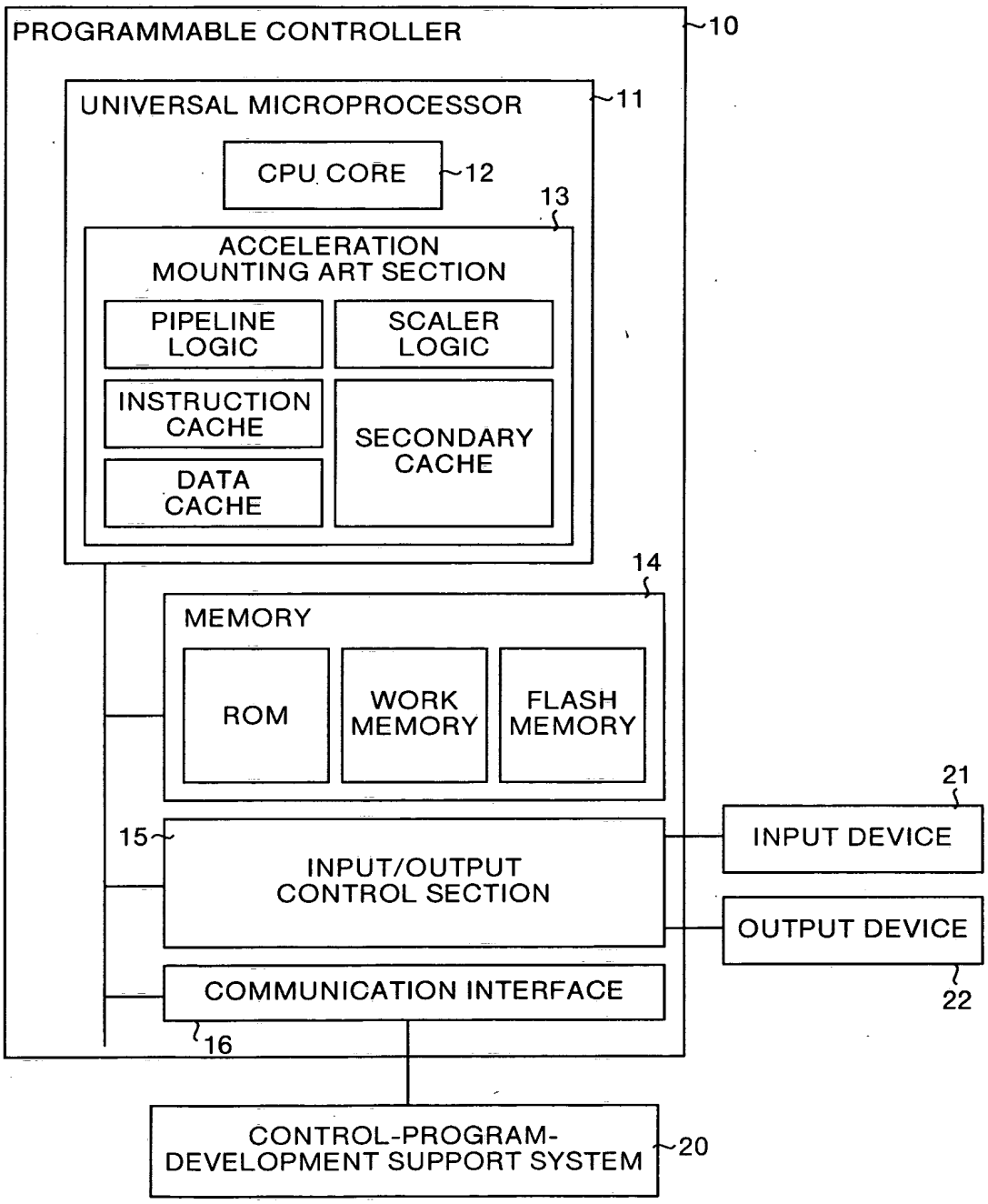

PROGRAMMABLE CONTROLLER
10
UNIVERSAL MICROPROCESSOR
11
CPU CORE
12
13
ACCELERATION MOUNTING ART SECTION
PIPELINE LOGIC
SCALER LOGIC
INSTRUCTION CACHE
SECONDARY CACHE
DATA CACHE
14
MEMORY
ROM
WORK MEMORY
FLASH MEMORY
21
15
INPUT/OUTPUT CONTROL SECTION
INPUT DEVICE
OUTPUT DEVICE
22
COMMUNICATION INTERFACE
16
CONTROL-PROGRAM-DEVELOPMENT SUPPORT SYSTEM
20

FIG.2

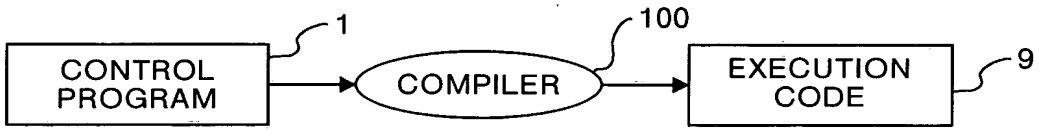

CONTROL PROGRAM
1
COMPILER
100
EXECUTION CODE
9

FIG.3

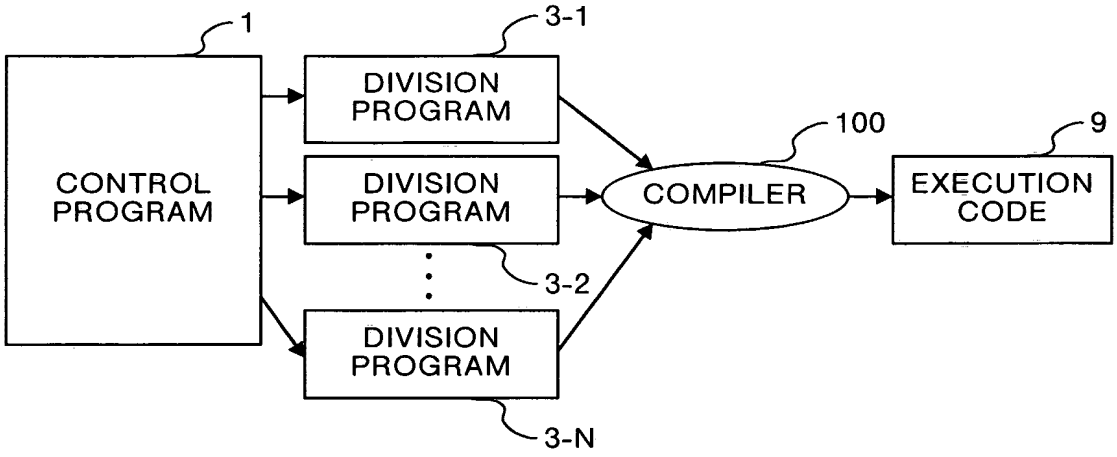

CONTROL PROGRAM
1
DIVISION PROGRAM
3-1
DIVISION PROGRAM
DIVISION PROGRAM
3-2
DIVISION PROGRAM
3-N
COMPILER
100
EXECUTION CODE
9

FIG.4

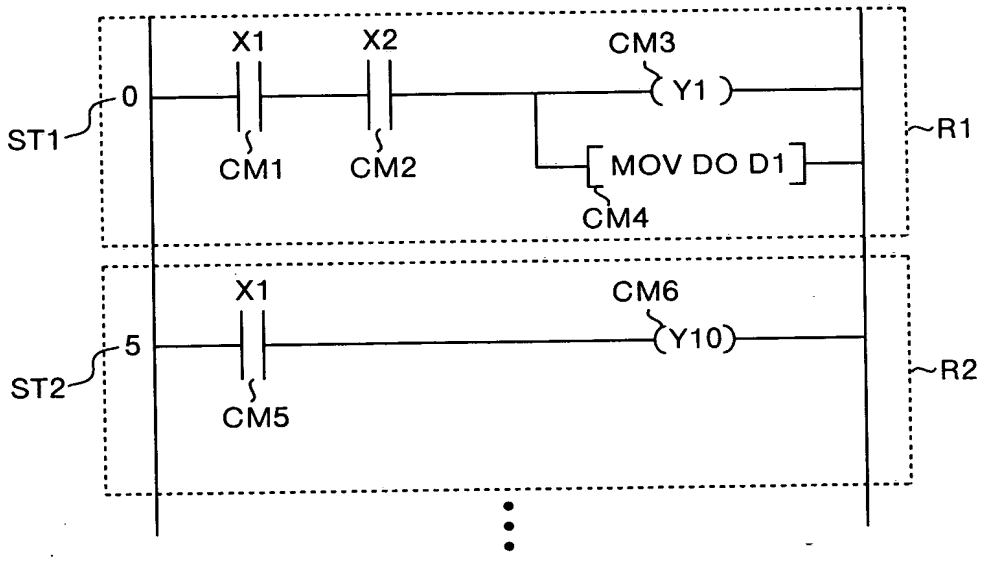


FIG.5

NUMBER OF STEPS	FILE NAME AFTER DIVISION
0~99	Control1.il
100~199	Control2.il
•	•
•	•
•	•

FIG.6

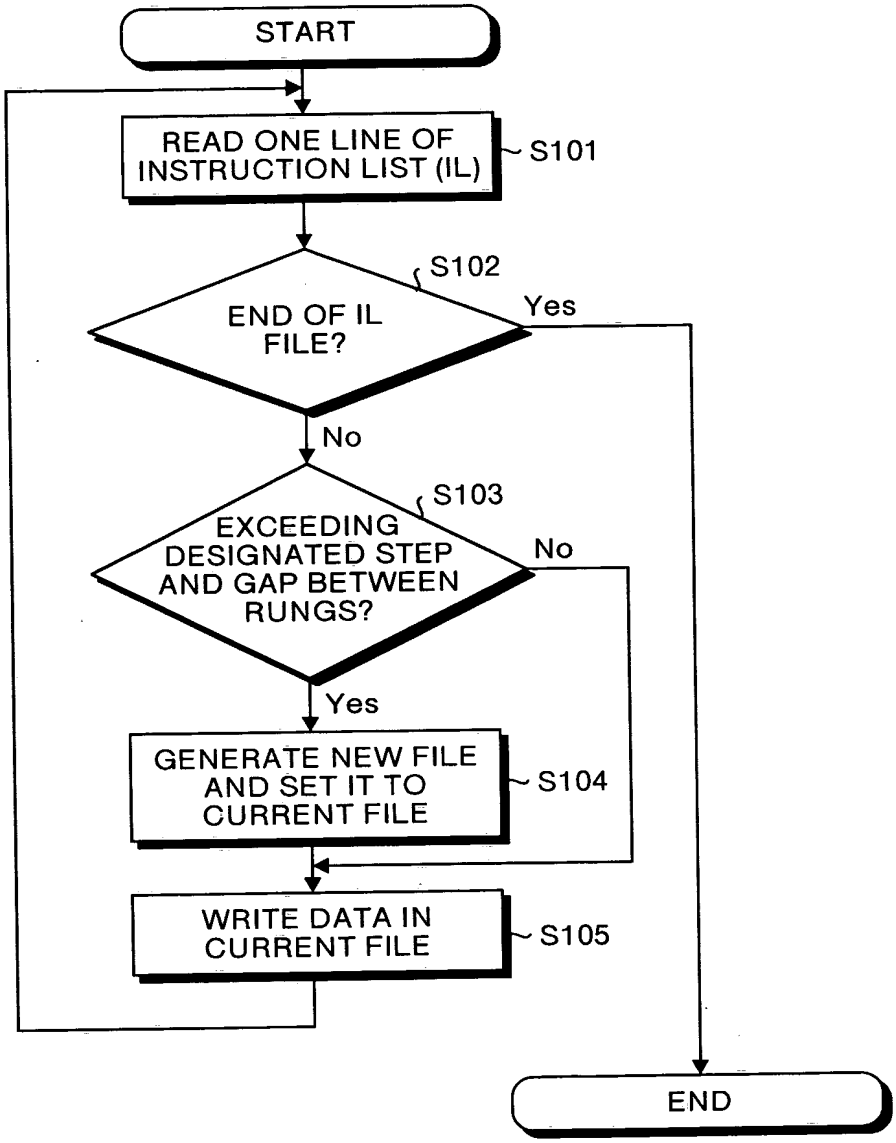

START
READ ONE LINE OF INSTRUCTION LIST (IL)
S101
S102
END OF IL FILE?
Yes
No
S103
EXCEEDING DESIGNATED STEP AND GAP BETWEEN RUNGS?
No
Yes
GENERATE NEW FILE AND SET IT TO CURRENT FILE
S104
WRITE DATA IN CURRENT FILE
S105
END

FIG.7

JUMP-DESTINATION LABEL NAME	FILE NAME AFTER DIVISION
P 1	Control1.il
P 2	Control2.il
•	•
•	•
•	•

FIG.8

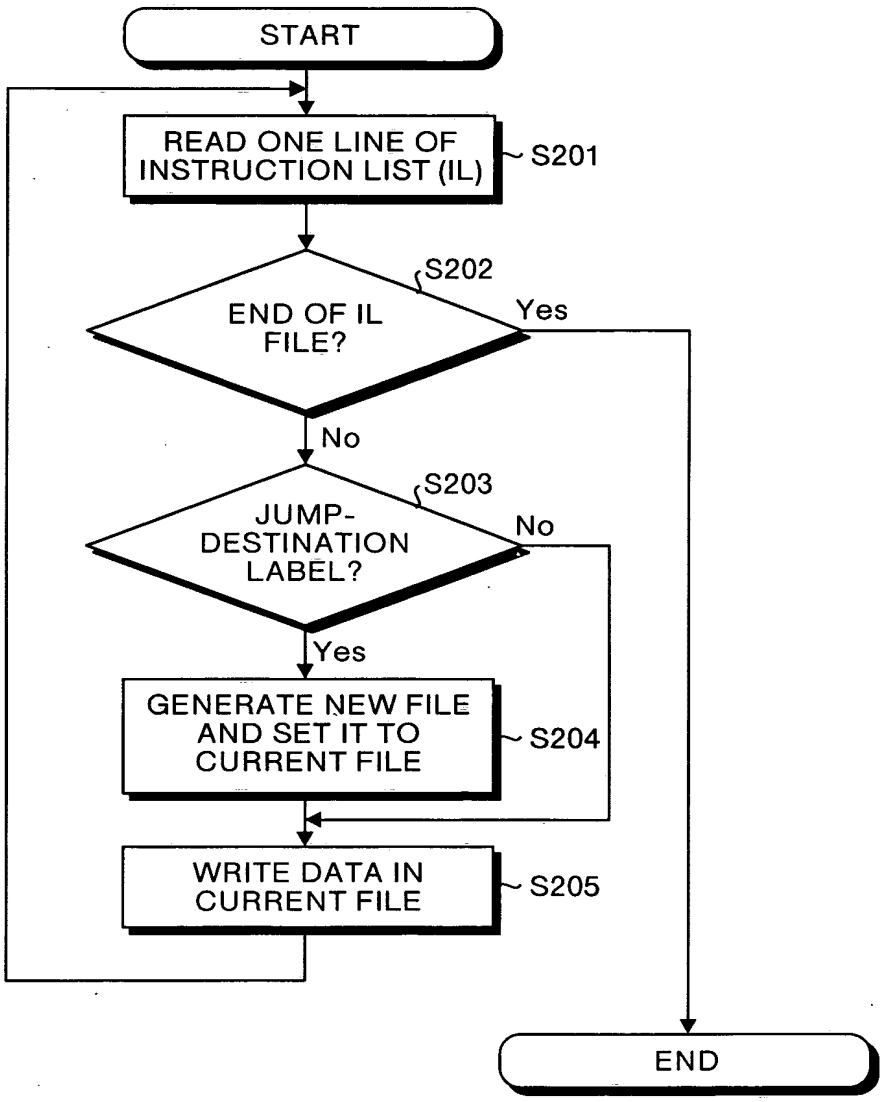

START
READ ONE LINE OF INSTRUCTION LIST (IL)
S201
S202
END OF IL FILE?
Yes
No
S203
JUMP-DESTINATION LABEL?
No
Yes
GENERATE NEW FILE AND SET IT TO CURRENT FILE
S204
WRITE DATA IN CURRENT FILE
S205
END

FIG.9

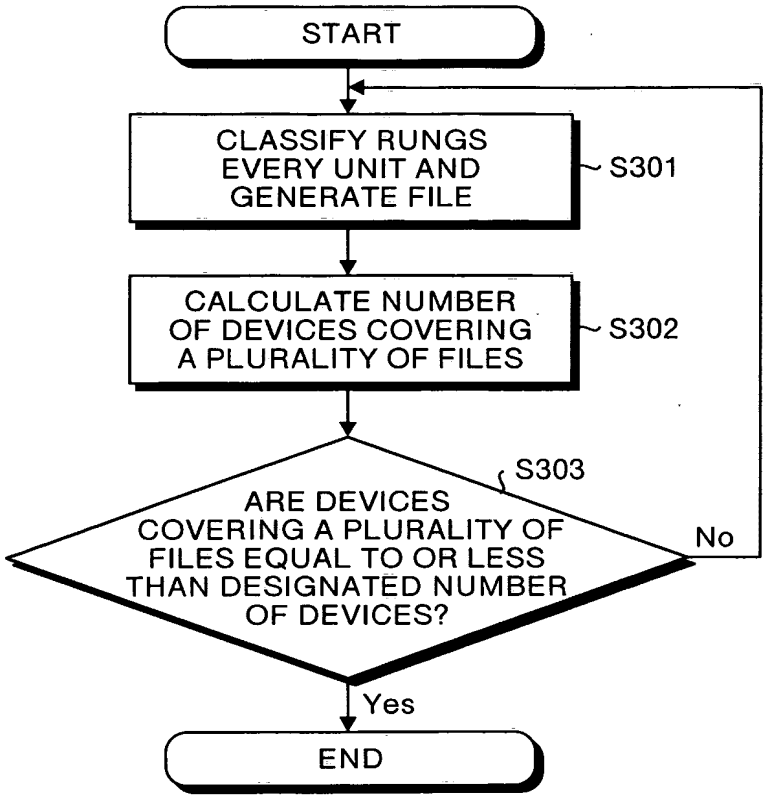

START
CLASSIFY RUNGS EVERY UNIT AND GENERATE FILE
S301
CALCULATE NUMBER OF DEVICES COVERING A PLURALITY OF FILES
S302
S303
ARE DEVICES COVERING A PLURALITY OF FILES EQUAL TO OR LESS THAN DESIGNATED NUMBER OF DEVICES?
No
Yes
END

FIG.10

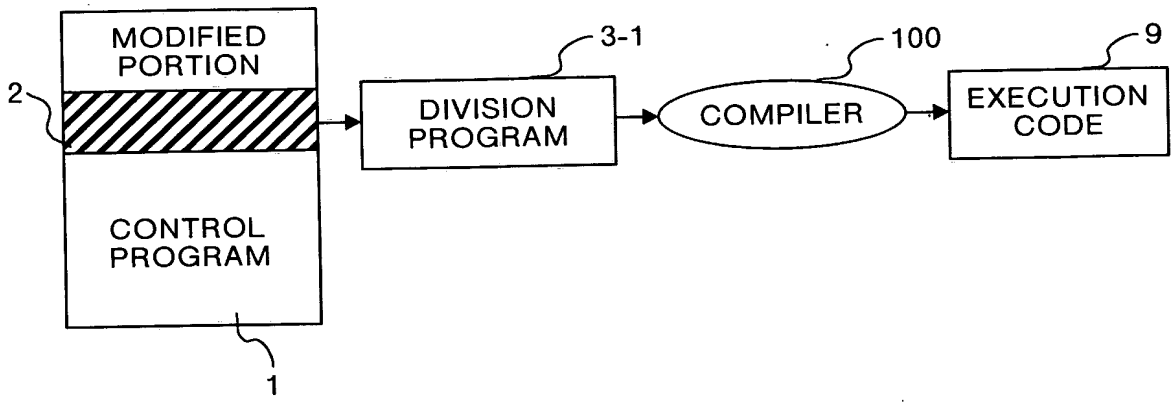

MODIFIED PORTION
2
CONTROL PROGRAM
1
3-1
DIVISION PROGRAM
100
COMPILER
9
EXECUTION CODE

FIG.11

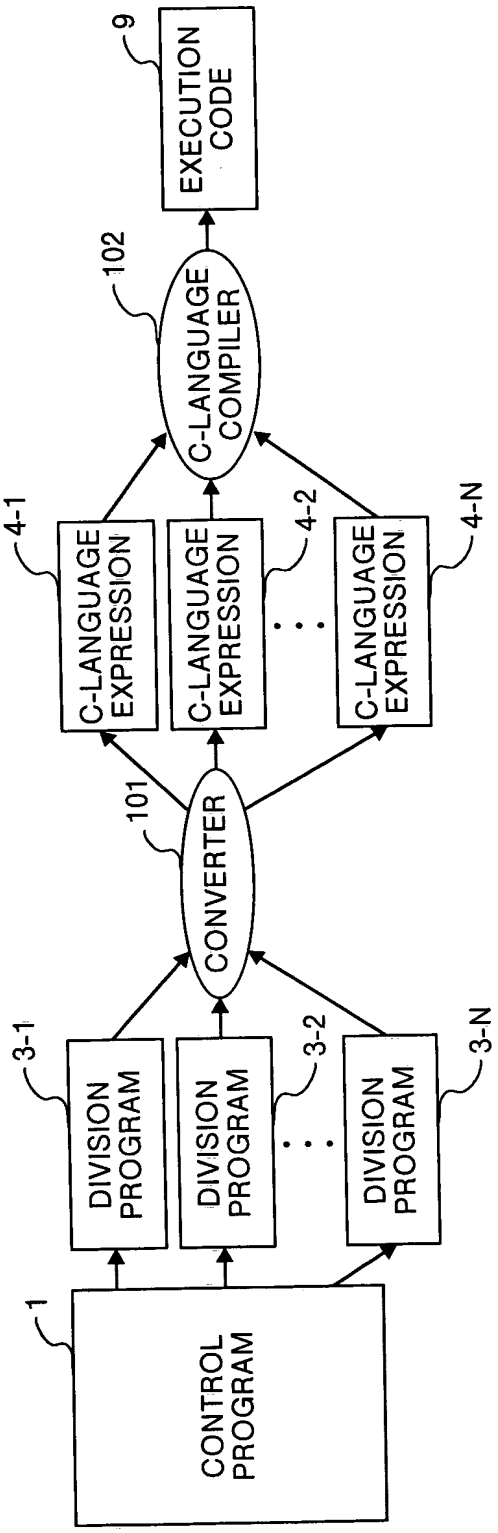

CONTROL PROGRAM
1
DIVISION PROGRAM
3-1
DIVISION PROGRAM
3-2
DIVISION PROGRAM
3-N
CONVERTER
101
C-LANGUAGE EXPRESSION
4-1
C-LANGUAGE EXPRESSION
4-2
C-LANGUAGE EXPRESSION
4-N
C-LANGUAGE COMPILER
102
EXECUTION CODE
9

FIG.12

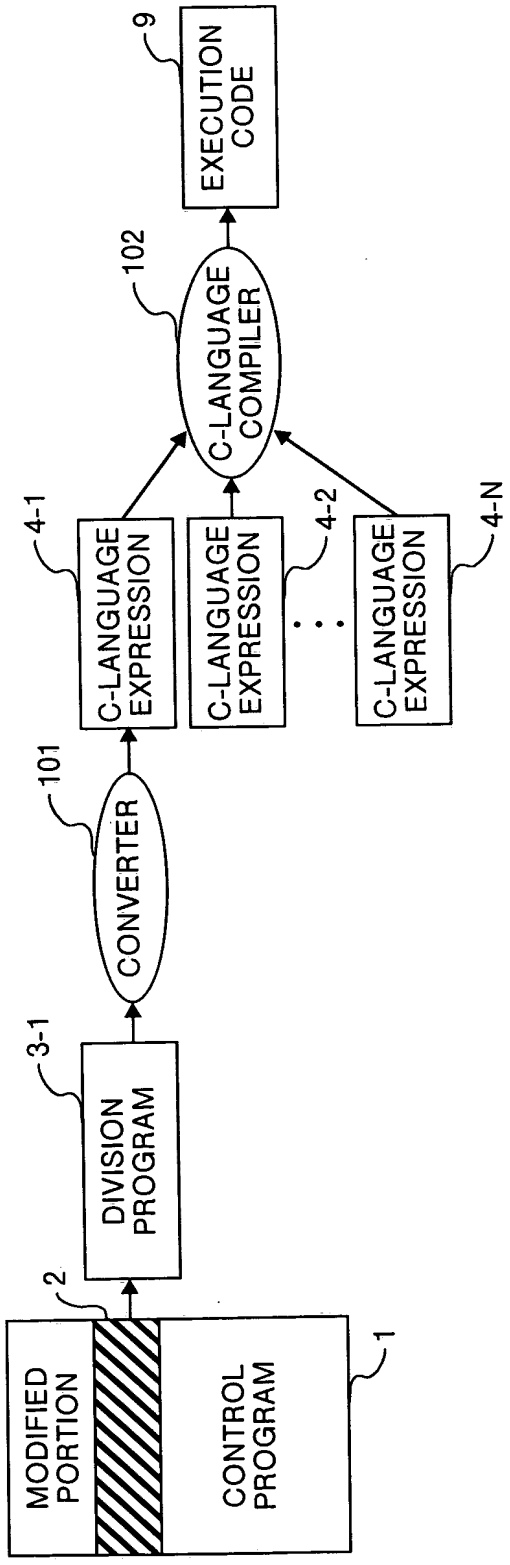

MODIFIED PORTION
CONTROL PROGRAM
1
2
DIVISION PROGRAM
3-1
CONVERTER
101
C-LANGUAGE EXPRESSION
4-1
C-LANGUAGE EXPRESSION
4-2
C-LANGUAGE EXPRESSION
4-N
C-LANGUAGE COMPILER
102
EXECUTION CODE
9

FIG.13

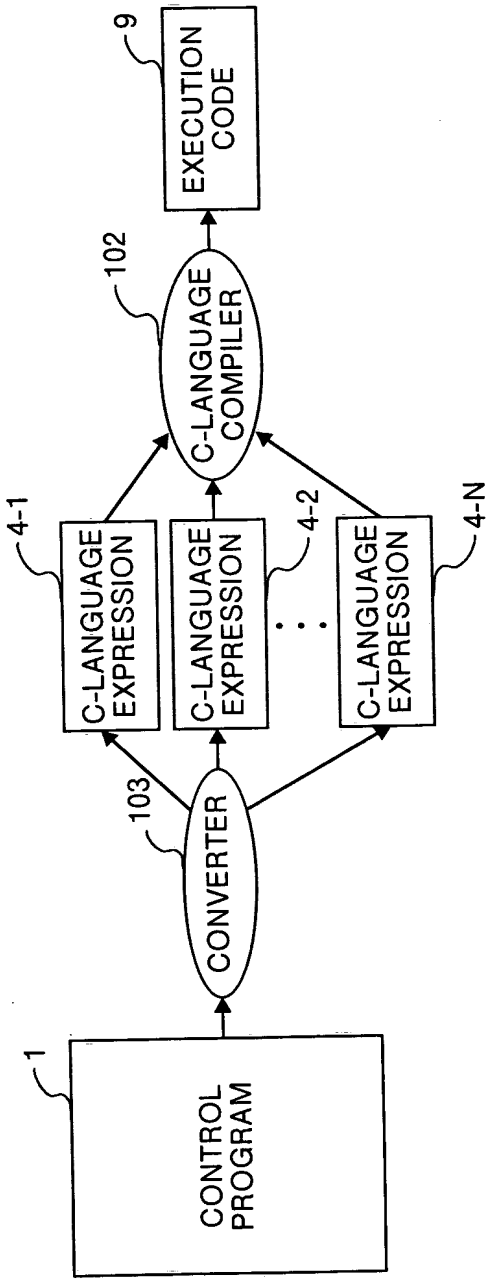

1
CONTROL PROGRAM
103
CONVERTER
4-1
C-LANGUAGE EXPRESSION
4-2
C-LANGUAGE EXPRESSION
4-N
C-LANGUAGE EXPRESSION
102
C-LANGUAGE COMPILER
9
EXECUTION CODE

FIG.14

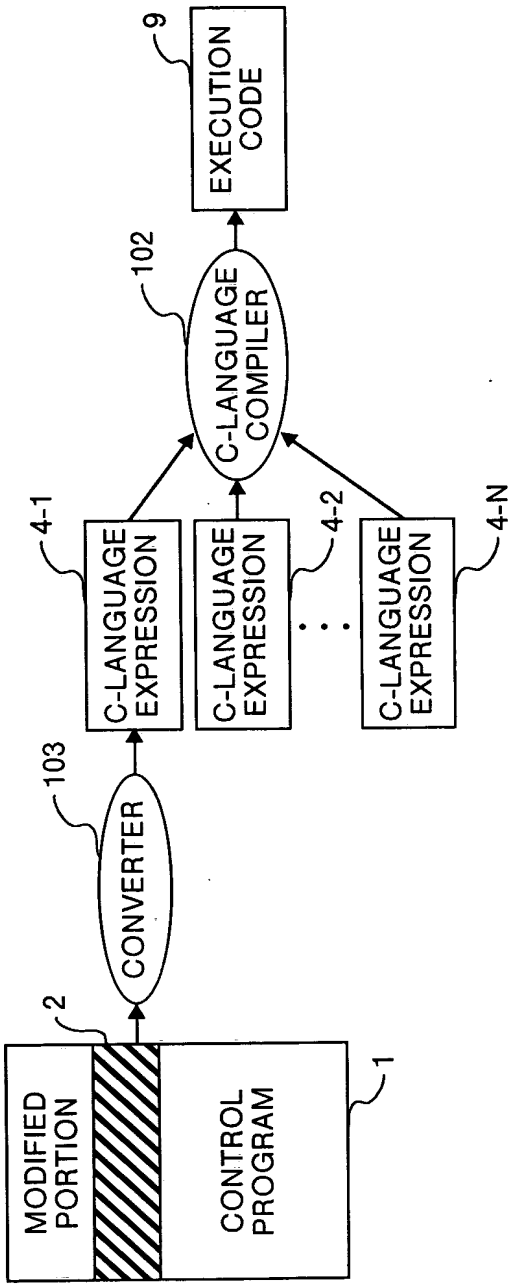

MODIFIED PORTION
CONTROL PROGRAM
CONVERTER
C-LANGUAGE EXPRESSION
C-LANGUAGE EXPRESSION
C-LANGUAGE EXPRESSION
C-LANGUAGE COMPILER
EXECUTION CODE
1
2
103
4-1
4-2
4-N
102
9

FIG.15

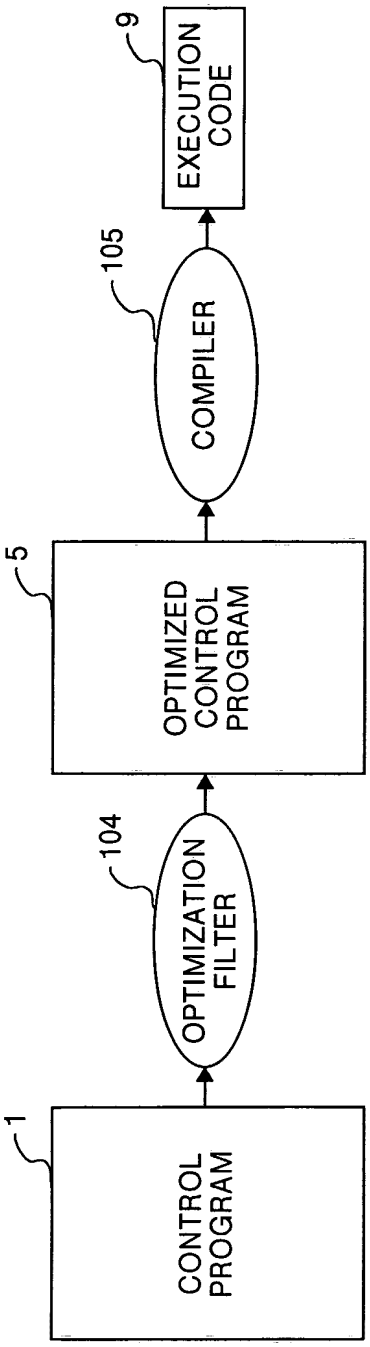

1
CONTROL PROGRAM
104
OPTIMIZATION FILTER
5
OPTIMIZED CONTROL PROGRAM
105
COMPILER
9
EXECUTION CODE

FIG.16

INSTRUCTION	SIZE (NUMBER OF STEPS)	PROCESSING TIME (msec)
LD,OUT	1K	XX
LD,AND,OUT	1K	YY
⋮	⋮	⋮
LDI,AND,OUT	10K	ZZ
⋮	⋮	⋮

FIG.17

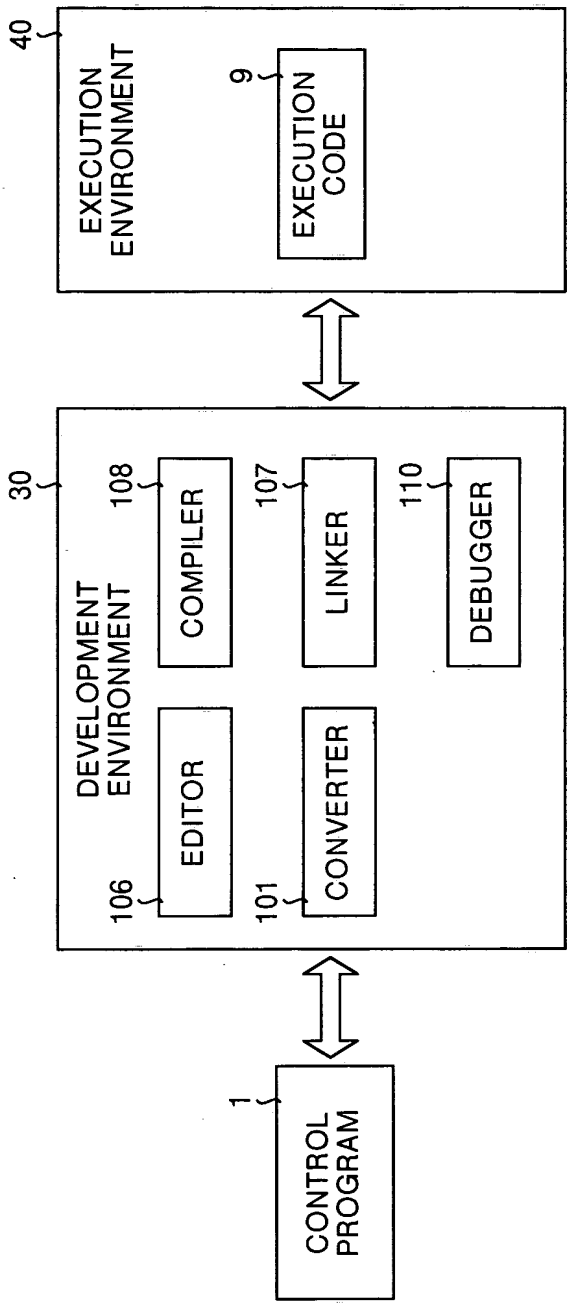

40
EXECUTION ENVIRONMENT
9
EXECUTION CODE
30
DEVELOPMENT ENVIRONMENT
108
COMPILER
107
LINKER
110
DEBUGGER
106
EDITOR
101
CONVERTER
1
CONTROL PROGRAM

FIG.18

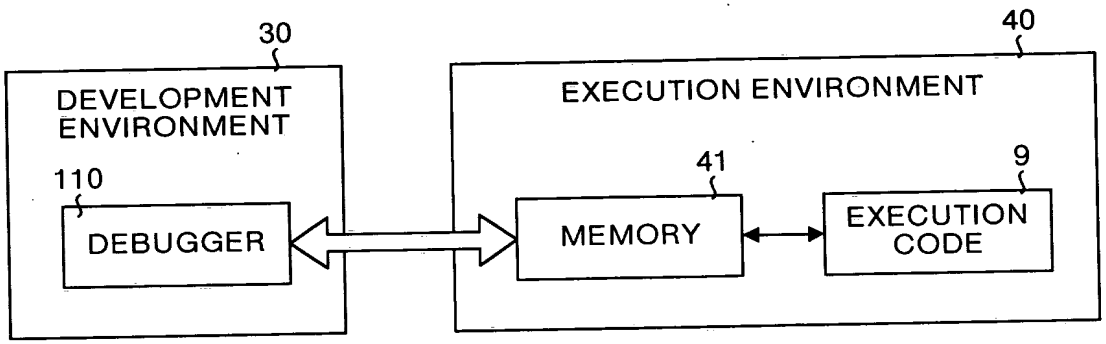

30
40
DEVELOPMENT ENVIRONMENT
110
DEBUGGER
EXECUTION ENVIRONMENT
41
MEMORY
9
EXECUTION CODE

FIG.19

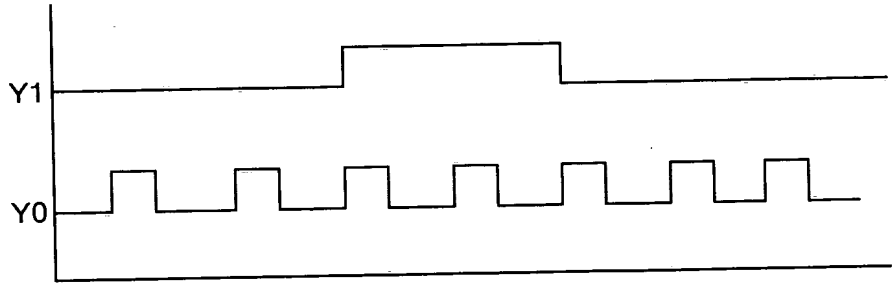

Y1
Y0

FIG.20

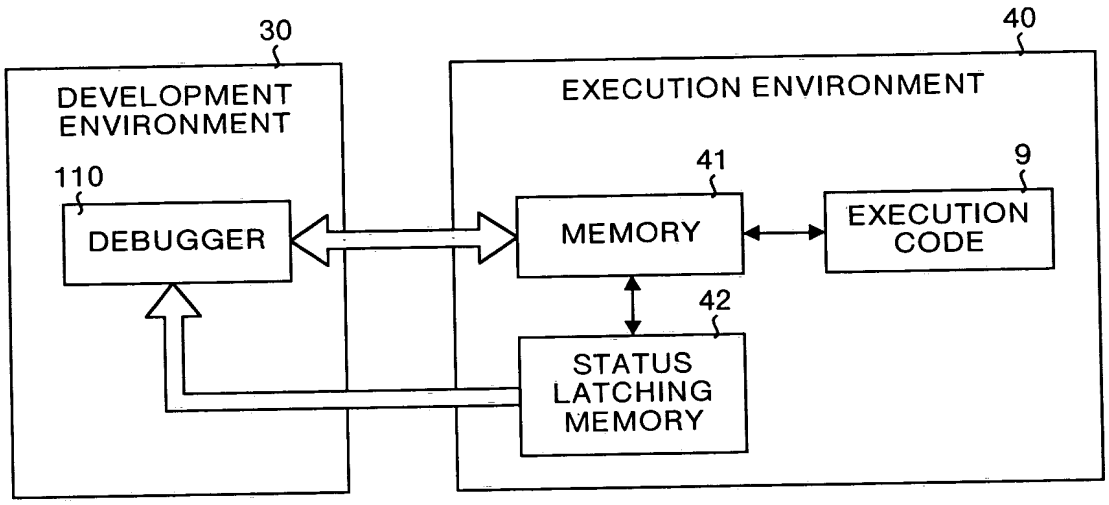

30
40
DEVELOPMENT ENVIRONMENT
EXECUTION ENVIRONMENT
110
DEBUGGER
41
MEMORY
9
EXECUTION CODE
42
STATUS LATCHING MEMORY

FIG.21

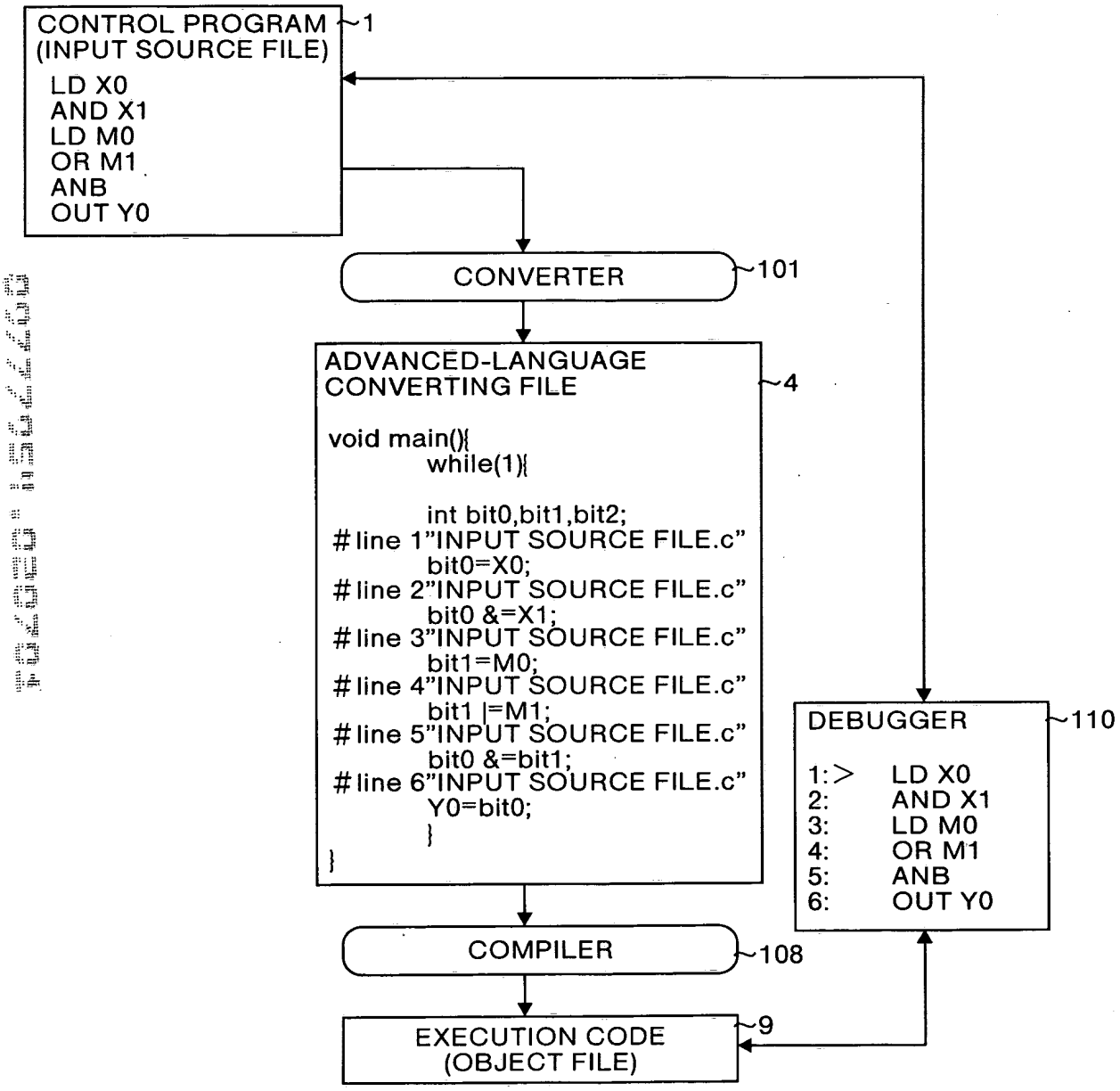

CONTROL PROGRAM
(INPUT SOURCE FILE)
LD X0
AND X1
LD M0
OR M1
ANB
OUT Y0
1
CONVERTER
101
ADVANCED-LANGUAGE
CONVERTING FILE
4
void main(){
while(1){
int bit0,bit1,bit2;
#line 1"INPUT SOURCE FILE.c"
bit0=X0;
#line 2"INPUT SOURCE FILE.c"
bit0 &=X1;
#line 3"INPUT SOURCE FILE.c"
bit1=M0;
#line 4"INPUT SOURCE FILE.c"
bit1 |=M1;
#line 5"INPUT SOURCE FILE.c"
bit0 &=bit1;
#line 6"INPUT SOURCE FILE.c"
Y0=bit0;
}
}
DEBUGGER
110
1:> LD X0
2: AND X1
3: LD M0
4: OR M1
5: ANB
6: OUT Y0
COMPILER
108
EXECUTION CODE
(OBJECT FILE)
9

FIG.22A

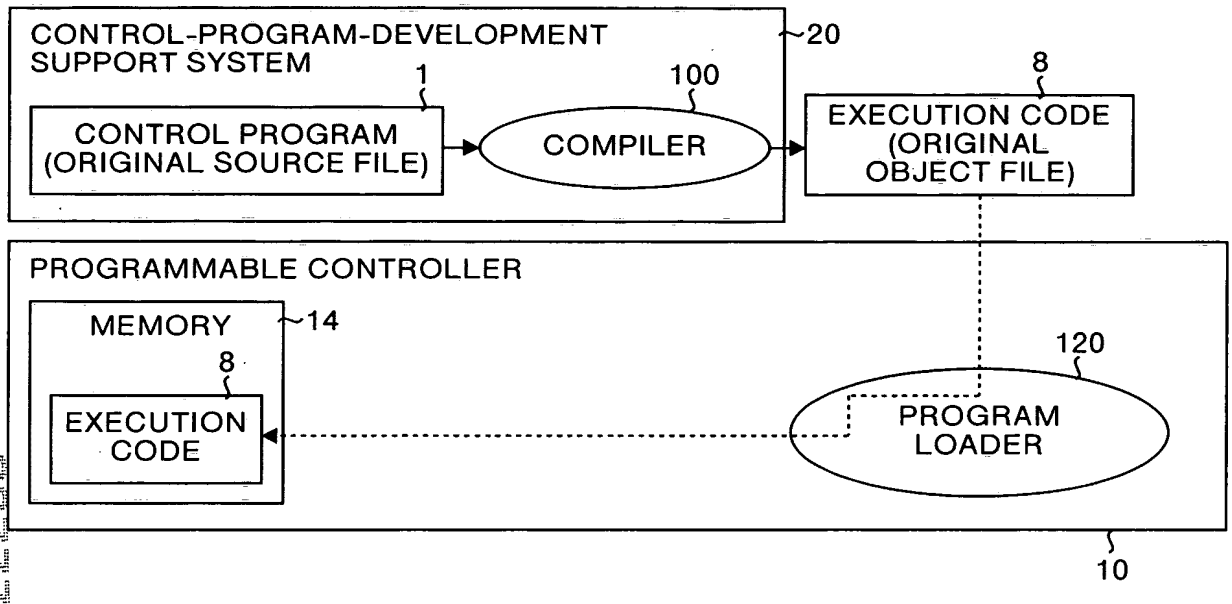


FIG.22B

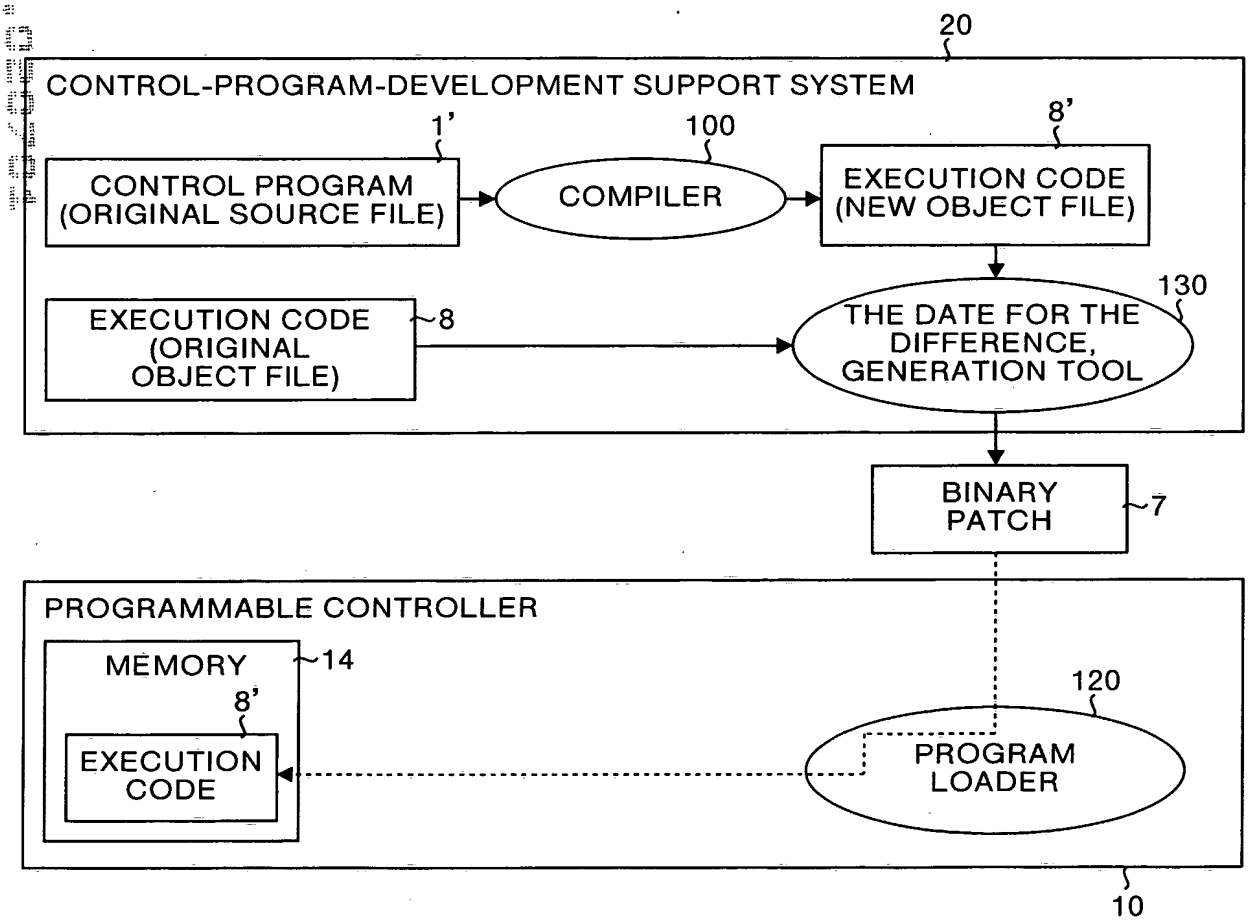


FIG.23

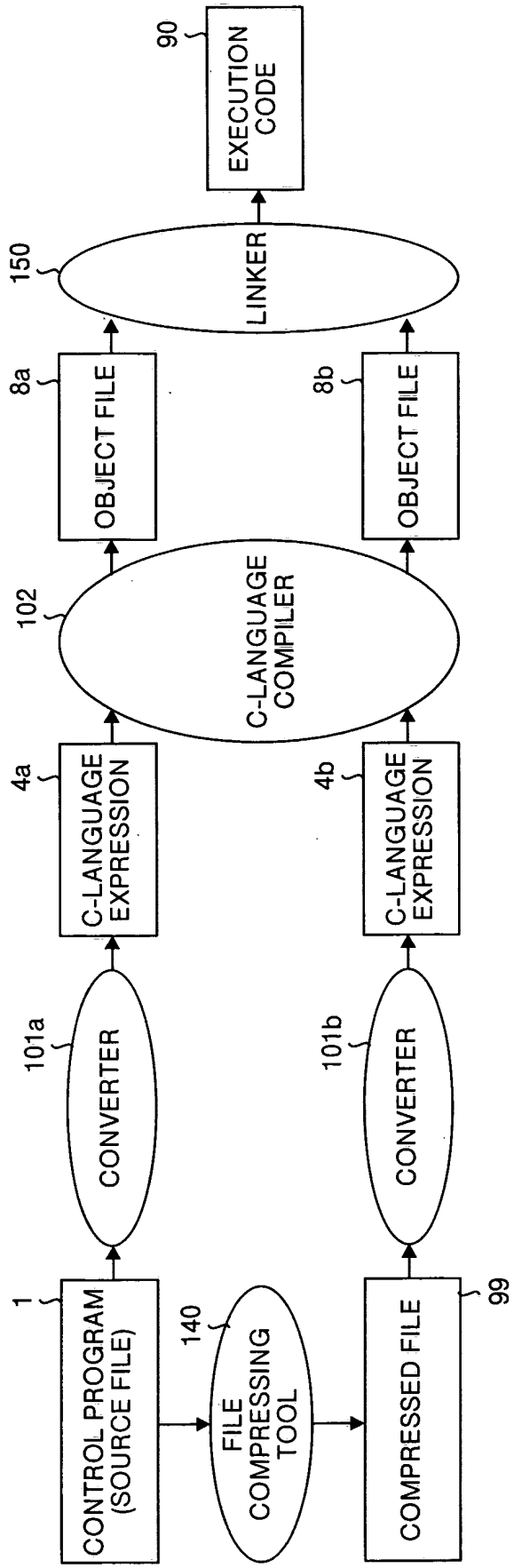

90
EXECUTION CODE
150
LINKER
8a
OBJECT FILE
8b
OBJECT FILE
102
C-LANGUAGE COMPILER
4a
C-LANGUAGE EXPRESSION
4b
C-LANGUAGE EXPRESSION
101a
CONVERTER
101b
CONVERTER
1
CONTROL PROGRAM (SOURCE FILE)
140
FILE COMPRESSING TOOL
99
COMPRESSED FILE

FIG.24

SOURCE FILE

```
LD      X0
OR      X2
LD      X1
OR      X3
ANB
LD      X4
AND     X5
ORB
OUT     M0
LD      X1C
MPS
AND     M8
OUT     Y30
MPP
OUT     Y31
LD      X1D
MPS
AND     M9
MPS
AND     M68
OUT     Y32
MPP
AND     T0
OUT     Y33
MPP
OUT     Y34
```

FIG.25

C-LANGUAGE EXPRESSION OF COMPRESSED FILE

```
signed char* _InputSourceName="sample.il.gz";

unsigned char _InputSource[]={
   0x1f,0x8b,0x08,0x08,0x9b,0xd9,0x84,0x39,0x00,0x03,0x73,0x61,0x6d,0x70,0x6c,0x65,
   0x2e,0x69,0x6c,0x00,0xf3,0x71,0xe1,0x8c,0x30,0xe0,0xe5,0xf2,0x0f,0xe2,0x8c,0x30,
   0xe2,0xe5,0xf2,0x01,0xf2,0x0c,0x21,0x3c,0x63,0x5e,0x2e,0x47,0x8f,0x27,0x88,0x90,
   0x09,0x88,0x0d,0xa4,0x4d,0x41,0x52,0x40,0x31,0xff,0xd0,0x10,0x4e,0x5f,0x03,0xa8,
   0x72,0x67,0x5e,0x2e,0xdf,0x80,0x60,0x88,0x0a,0x5f,0x0b,0x88,0x64,0xa4,0xb1,0x01,
   0x48,0x34,0x00,0xc6,0x33,0x84,0xaa,0x75,0x41,0x56,0x6b,0x89,0xcc,0x31,0x83,0xeb,
   0x34,0x82,0xea,0x04,0x09,0x87,0x18,0xc0,0x44,0x8d,0x51,0xcc,0x03,0x3a,0x08,0x00,
   0x10,0xeb,0x7d,0x3a,0xb9,0x00,0x00,0x00};
```

FIG.26

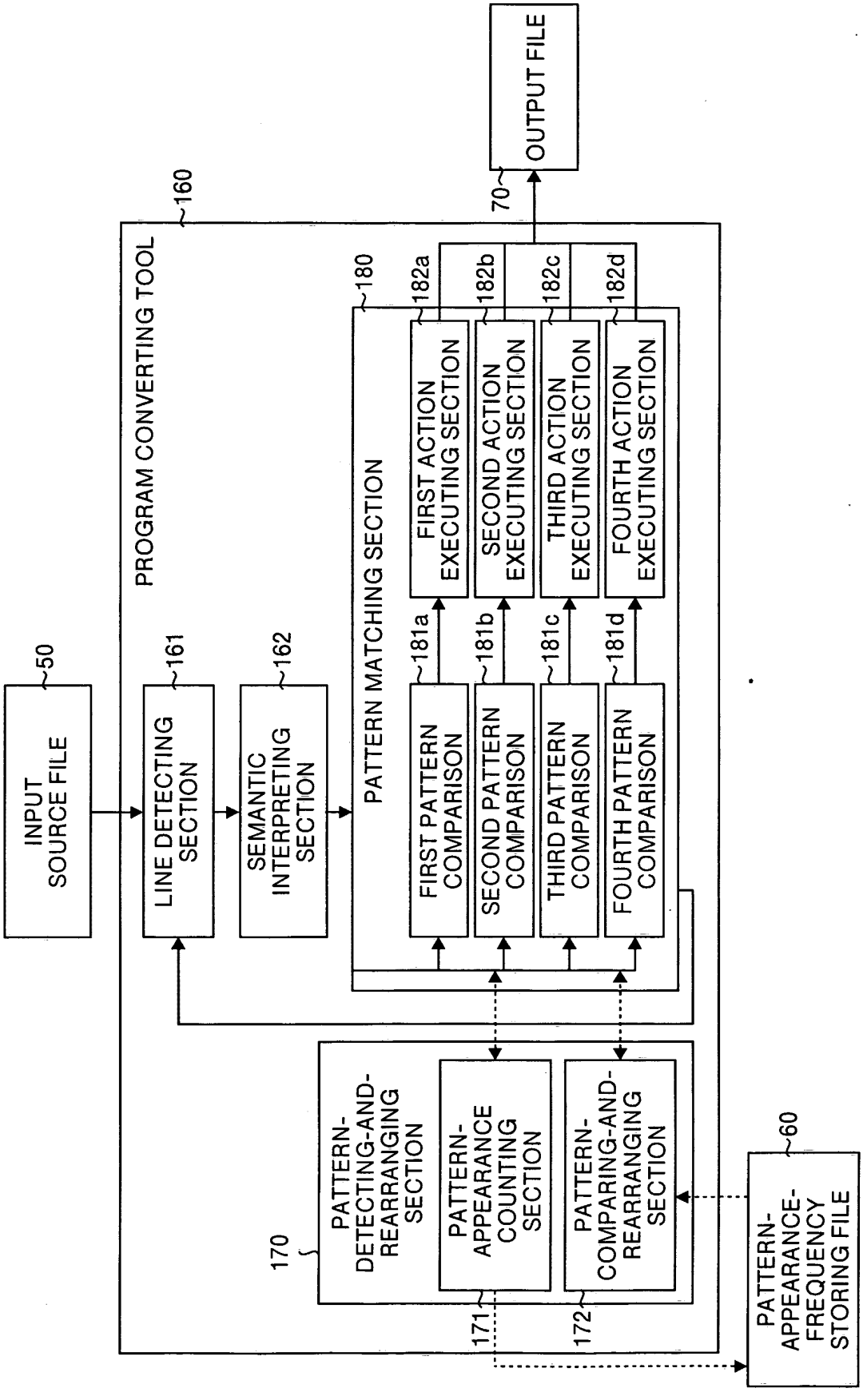
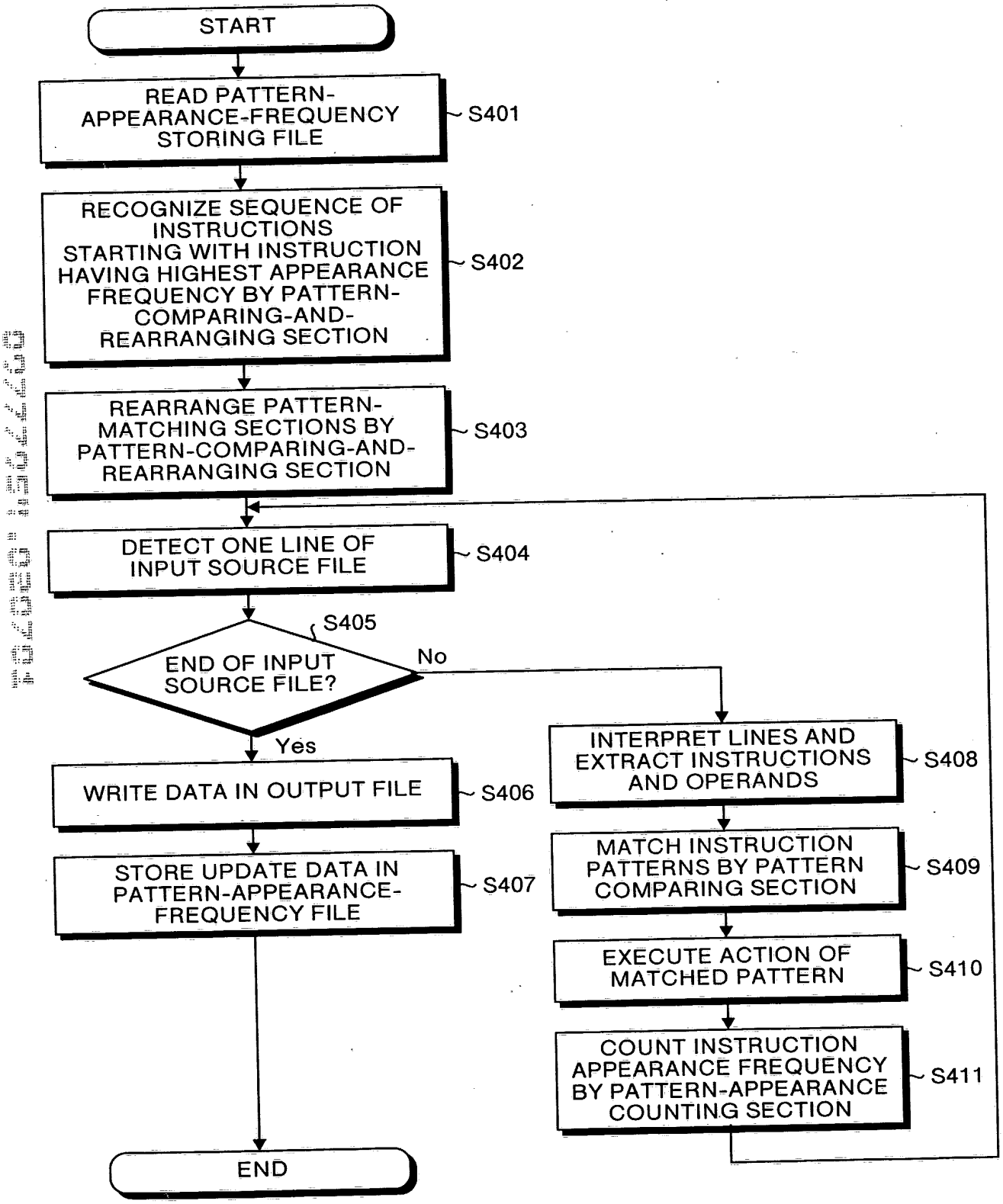

INPUT SOURCE FILE
50
LINE DETECTING SECTION
161
SEMANTIC INTERPRETING SECTION
162
PROGRAM CONVERTING TOOL
160
PATTERN MATCHING SECTION
180
FIRST PATTERN COMPARISON
181a
SECOND PATTERN COMPARISON
181b
THIRD PATTERN COMPARISON
181c
FOURTH PATTERN COMPARISON
181d
FIRST ACTION EXECUTING SECTION
182a
SECOND ACTION EXECUTING SECTION
182b
THIRD ACTION EXECUTING SECTION
182c
FOURTH ACTION EXECUTING SECTION
182d
OUTPUT FILE
70
PATTERN-DETECTING-AND-REARRANGING SECTION
170
PATTERN-APPEARANCE COUNTING SECTION
171
PATTERN-COMPARING-AND-REARRANGING SECTION
172
PATTERN-APPEARANCE-FREQUENCY STORING FILE
60

FIG.27


START
READ PATTERN-APPEARANCE-FREQUENCY STORING FILE
S401
RECOGNIZE SEQUENCE OF INSTRUCTIONS STARTING WITH INSTRUCTION HAVING HIGHEST APPEARANCE FREQUENCY BY PATTERN-COMPARING-AND-REARRANGING SECTION
S402
REARRANGE PATTERN-MATCHING SECTIONS BY PATTERN-COMPARING-AND-REARRANGING SECTION
S403
DETECT ONE LINE OF INPUT SOURCE FILE
S404
S405
END OF INPUT SOURCE FILE?
No
Yes
WRITE DATA IN OUTPUT FILE
S406
STORE UPDATE DATA IN PATTERN-APPEARANCE-FREQUENCY FILE
S407
END
INTERPRET LINES AND EXTRACT INSTRUCTIONS AND OPERANDS
S408
MATCH INSTRUCTION PATTERNS BY PATTERN COMPARING SECTION
S409
EXECUTE ACTION OF MATCHED PATTERN
S410
COUNT INSTRUCTION APPEARANCE FREQUENCY BY PATTERN-APPEARANCE COUNTING SECTION
S411